

REMARKS

Reconsideration and further examination of this application is respectfully requested. Claims 4-6 and 9 were previously cancelled. Claims 1, 7, 10, 11, 14, and 17 were previously amended. Claims 1, 7, 14 and 17 have been further amended to clarify the meaning of the claims. Claims 2, 3, 8, 12, 13, 15, and 16 are presented in original form. Therefore, Applicants submit claims 1-3, 7, 8 and 10-17 for further examination.

Claims 1, 7, 14 and 17 have been amended to include a limitation for the multiple-path driver such that “said multiple-path driver function[s] without affecting said computer operating system operation and said multiple-path driver [is] provided without modifying said computer operating system source code.” Support for the multiple-path driver functioning without affecting the computer operating system may be found in the original specification of the subject patent application on page 5 at lines 18-19, which state: “After the operating system is functional within a computer system, the multiple-path driver may be loaded into the computer system so as not to affect the operating system.” Support for the multiple-path driver being provided without modifying the computer operating system source code may be found in the original specification of the subject patent application on page 2 at lines 18-21, which state: “The multiple-path driver creates virtual (or non-physical) data paths within a single computer operating system without modifying the source code of the operating system to provide increased throughput and minimize the effect of failed data paths.” Claim 1 has also been amended to move a limitation to “inserting [the] multiple-path driver between driver stack levels” from the preamble into the body of the claim in order to better clarify the limitations of the claim.

No new matter has been added by the above described amendments.

In the subject Office Action, the Examiner rejected claims 1-3, 7-8 and 10-17 under 35 U.S.C. 103(a) as being unpatentable over Pooni et al. (US PGPub No. 2004/0225764) (hereinafter Pooni) in view of Padmanabhan et al. (US PGPub No. 2004/0107300) (hereinafter Padmanabhan).

In the rejection for claim 1, the Examiner asserted that paragraphs 0012-0014 of Pooni disclose “selectively intercepting device commands from upper-level drivers of said operating system of said host computer,” as recited in claim 1 of the subject patent application. Applicants fail to find support for disclosure of intercepting device commands from upper-level drivers in paragraphs 0012-0014 of Pooni. Paragraphs 0012-0014 of Pooni state:

“[0012] FIG. 1 illustrates a driver architecture for the Linux SCSI I/O Subsystem. The Linux SCSI I/O subsystem (hereinafter SCSI subsystem 1000) has a three (3) level driver architecture with the "upper" level layer 100 being closest to a user space/kernel space interface 110, while the "lower" level layer 150 is closest to hardware (e.g., physical SCSI device, not shown). As will be explained further below, the upper level drivers (ULDs) in upper level 100 are commonly known by a terse two letter abbreviation (e.g., "sd" for SCSI disk driver). The names of the corresponding module drivers which, for historical reasons, sometimes differ from the built-in driver names are shown in brackets in FIG. 1.

[0013] The ULDs in upper level layer 100 maintain the kernel side of the Linux OS's user-kernel interface for the logical class of SCSI devices they represent (e.g., disks). They are also responsible for managing certain kernel and SCSI subsystem 1000 resources such as kernel memory and SCSI command structures. Applications in the user space access these ULDs by opening a device file (block or char) typically found in the /dev directory tree.

[0014] Accordingly, the upper level layer 100 supports the kernel/user interface. For example, in the case of sd (disk) drivers and sr (CD-ROM) drivers, the kernel/user interface is a block device interface, while for st (tape) drivers and sg (generic) drivers, the kernel/user interface is a character device interface. Any operation using the SCSI subsystem 1000 (e.g., reading a sector from a disk) involves one driver at each of the 3 levels, e.g., sd drivers at upper level 100, SCSI mid-level driver at SCSI mid level layer 125 and an HBA driver of lower level layer 150.”

Paragraphs 0012-0014 are found in the “Background” section of the Pooni reference and appear to disclose the standard operation of the upper level drivers when multiple physical communication paths are available for a SCSI device to communicate with a computer. In standard operation, the device commands from the upper-level drivers of the operating system are not intercepted and, accordingly, Applicants cannot locate a reference to intercepting device commands from upper-level drivers in the passages indicated by the Examiner. Further, Applicants cannot locate a reference to intercepting device commands from upper-level device drivers in the remainder of the Pooni reference. Thus, Pooni fails to disclose, teach, or suggest the limitation of claim 1 of the subject patent application for “selectively intercepting device commands from upper-level drivers of said operating system of said host computer.”

Also in the rejection for claim 1, the Examiner asserted that Fig. 5 and paragraphs 0065-0067 and 0071 disclose “selectively presenting said computer data storage devices to said upper-level drivers of said operating system of said host computer as a single virtual computer storage

device,” as recited in claim 1 of the subject patent application. Applicants fail to find a disclosure in Fig. 5 or paragraphs 0065-0067 and 0071 that discloses presenting the data storage devices to the upper level device drivers of the operating system as a single virtual computer storage device. Applicants understanding of the Pooni reference is that the system of Pooni provides a means to identify devices with multiple physical communications paths and present device identification strings for the multiple communications paths in a manner that permits a user or user program (daemon) to identify devices with multiple communications paths by identifying each communication path to the user or user program with a device identification scheme that highlights that the listed devices are actually the same device, but with different physical communications paths. For instance, at paragraph 0050 Pooni states: “At least one embodiment of the present invention provides a method by which a user mode application obtains all physical paths that point to a logical unit on a newly-discovered small computer interface (SCIS) device.” *Emphasis added.* Further, at paragraph 0065 Pooni states: “FIG. 5 is a sequence diagram illustrating a method of identifying multiple paths to a single device.”

Emphasis added. Thus, Pooni discloses a system that obtains and/or identifies multiple paths to a single logical unit device, but identifying and/or obtaining multiple paths for a logical unit device does not entail presenting the multiple paths to the upper level drivers as a single virtual computer storage device. The Pooni system presents the multiple paths of the logical unit device to the user or user program, not the upper level device drivers, as is recited in claim 1 of the subject patent application. Further, the Pooni system does not present the multiple paths as a single virtual computer data storage device, but as multiple computer storage devices with device identifiers that permit the user or user program to identify that each of the presented devices actually refer to the same physical device. The example system disclosed by Pooni at paragraphs 0100-0109 shows that a single device with four potential communication paths will be presented to the user or user program via four different device identifiers (paragraphs 0105-0108). *See* Pooni at paragraphs 0100-0109 which state:

“[0100] In the following example, assume that one (1) UID, "123456789", and two h,b,t,l addresses (paths) for that UID, (0,1,0,2) and (0,3,4,2) were forwarded to the user mode daemon 118. The daemon 118 is going to ask the following to each existing, standard device file (these are /dev/sd[a-z]; /dev/sg[0-255]; /dev/st[0-32] and /dev/nst[0-32]):

[0101] (a) the device file's major number (this number tells the daemon 118 if the device

file is a disk, generic, or tape device file);

[0102] (b) the device file's minor number; and

[0103] (c) what is the device file's (h,b,t,l)?

[0104] Taking actual device file names, assume that /dev/sg5 (SCSI generic major number, minor number 5) and /dev/sdaa (SCSI disk major number, minor number 0, one partition) both pointed to h,b,t,l address (0,1,0,2); and /dev/sg1 and /dev/sdba both pointed to (0,3,4,2). The process would leave those files there, and create, in addition to them, the following filter-specific device files:

[0105] /dev/trdisk/123456789.sub.--0.sub.--0 (uid path partition, same major/minor as /dev/sdaa)

[0106] /dev/trdisk/123456789.sub.--1.sub.--0 (same major/minor as /dev/sdba)

[0107] /dev/trgen/123456789.sub.--0 (no partitions for generic devices, same major/minor as /dev/sg5)

[0108] /dev/trgen/123456789.sub.--1 (same major/minor as /dev/sg1)

[0109] Accordingly, /dev/trdisk/123456789.sub.--0.sub.--0 and /dev/trgen/123456789.sub.--0 both talk to the same device through the same path (since the UID and device path numbers match). Further, /dev/trdisk/123456789.sub.--1.sub.--0 and /dev/trgen/123456789.sub.--1 also refer to the same device, but from two different paths (since only the UID values match). This relationship is not at all obvious with the original files (e.g., /dev/sg5 and /dev/sdaa; /dev/sg1 and /dev/sdba). That is because the standard device file names offer no information indicating that they all point to the same LUN.” *Emphasis added.*

Thus, Pooni fails to disclose, teach, or suggest the limitation of claim 1 of the subject patent application for “selectively presenting said computer data storage devices to said upper-level drivers of said operating system of said host computer as a single virtual computer data storage device.”

Again, in the rejection for claim 1, the Examiner asserts that Pooni does not disclose “creating a virtual data path between said multiple-path drive and said computer data storage devices; and selectively directing device commands from said multiple-path driver to a virtual host bus adapter driver within said multiple-path along said virtual path,” as recited in claim 1 of the subject patent application. However, the Examiner also asserted that Padmanabhan

“discloses a virtual host controller interface with multipath input/output including data path and host bus adapter driver.” Paragraph 0028 of Padmanabhan states: “This application describes the new multiplexed I/O (MPXIO) architecture, along with a set of changes to the core of the OS which support this architecture.” The disclosure of Padmanabhan clearly intends that the core of the operating system is modified to support the system of Padmanabhan. If the operating system is not modified, the system disclosed by Padmanabhan would not be able to support the multiplexed I/O because of the low-level operations needed to control the system as disclosed by Padmanabhan. Therefore, the system of Padmanabhan would not be able to manage devices that are accessible through multiple physical communications paths without low-level access to the operating system. Thus, without modifying the core of the operating system, Padmanabhan would be unsatisfactory for its intended purpose of managing devices.

Similarly, the disclosure of Pooni also suggests that the system of Pooni requires changes to the operating system kernel code and has affects on the operation of the kernel of the operating system. For instance, paragraph 0067 of Pooni states:

“[0067] In accordance with the method and apparatus of the present invention, and as will be explained in further detail below, a filter driver 114 which is a kernel component that runs in the kernel space of the Linux OS, is inserted as a module between the upper level disk drivers (device drivers 108) and the mid-level SCSI driver 102 in the SCSI subsystem. The filter driver 114, in kernel space, will be used to create a filter-specific device file name, in user space, that may be used to differentiate all multiple paths to a single discovered device 106, such as to a single logical unit.” *Emphasis added.*

As with Padmanabhan, removing the filter driver as a component of the operating system would not permit the filter driver proper access to system resources as disclosed by the Pooni reference and the Pooni system would be unfit for its purpose of identifying multiple communications paths for a data storage device. Thus, without modifying the kernel source code and/or affecting operation of the kernel of the operating system, the Pooni system would be unsatisfactory for its intended purpose of identifying multiple physical communications paths for a data storage device.

As amended claim 1 of the subject patent application recites: “inserting a multiple-pat driver between driver-stack levels of an operating system operating on said host computer without affecting said operating system operation, said multiple path driver further being provided without modifying said operating system source code.” As amended claim 1 clearly

recites that the multiple-path driver operates without affecting the operation of the operating system and without modifying the operating system source code. Both Pooni and Padmanabhan disclose systems that modify the source code of the operating system and/or affect the operation of the operating system. Thus, both Pooni and Padmanabhan teach away from providing a system to manage multiple communications paths for devices without modifying the operating system source code and/or affecting the operation of the operating system. Further, as stated above, without modifying the source code and/or affecting the operation of the operating system, both Pooni and Padmanabhan are unsatisfactory for their intended purposes. Thus, the combination of Pooni and/or Padmanabhan with other references, including both Pooni and Padmanabhan, is improper under 35 U.S.C. 103(a).

For the above stated reasons, Pooni does not disclose, teach, or suggest the limitations of as amended claim 1 of the subject patent application for intercepting device commands from upper-level drivers, for presenting the computer data storage devices to the upper-level drivers as a single virtual data storage device, or for providing a multiple-path driver without modifying operating system source code and/or affecting operating system operation. Padmanabhan does not make up for the deficiencies of Pooni. Thus, the rejection of claim 1 of the subject patent application under 35 U.S.C. 103(a) fails as the combination of Pooni and Padmanabhan does not disclose all of the limitations the claim. Further, both Pooni and Padmanabhan disclose systems that teach the need for, and are unsatisfactory for their intended purposes without, modifying the operating system source code and/or affecting the operation of the operating system. Therefore, Pooni and Padmanabhan teach away from claim 1 of the subject patent application and the Examiner has failed to make a proper *prima facie* case for obviousness under 35 U.S.C. 103(a).

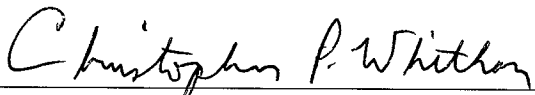
As amended independent claims 7, 14, and 17 have similar limitations to claim 1. Thus, the rejections of claims 7, 14, and 17 under 35 U.S.C. 103(a) also fail and are improper for the same reasons as set forth for claim 1. Claims 2, 3, 8, 10-13, 15, and 16 are dependent claims depending from independent parent claims 1, 7, or 14. Therefore, dependent claims 2, 3, 8, 10-13, 15, and 16 are also not rendered obvious under 35 U.S.C. 103(a) since the independent parent claims 1, 7, and 14 are not rendered obvious by the prior art.

The Examiner made certain references of record, but did not apply these references to the subject claimed invention. Applicants have reviewed these references and believe no further response is required.

For these reasons, claims 1-3, 7, 8 and 10-17 are proper and are considered to be patentable over the prior art. Therefore, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Dated this 15th day of January, 2008.

COCHRAN FREUND & YOUNG LLC

By: 
Christopher P. Whitham #53,769
Attorney for Applicants
2026 Caribou Drive, Suite 201
Fort Collins, CO 80525
(970) 492-1100
Fax: (970) 492-1101